

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 10-222525

(43)Date of publication of application : 21.08.1998

(51)Int.Cl.

G06F 17/30

G06F 12/00

G06F 13/00

(21)Application number : 09-024804

(71)Applicant : HITACHI LTD

(22)Date of filing : 07.02.1997

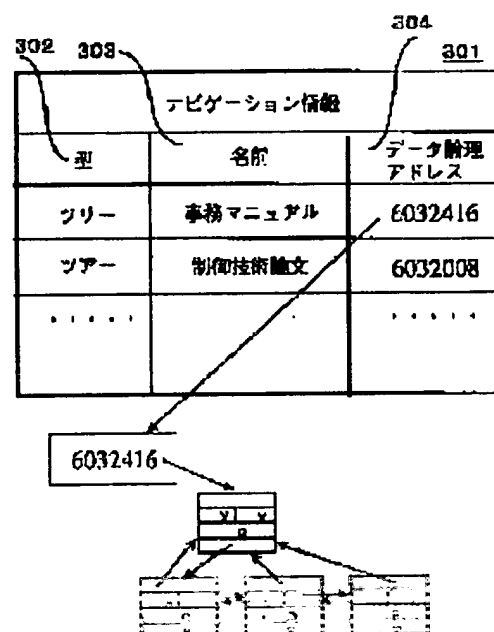
(72)Inventor : YAMADA TAKAAKI
KOJIMA HIROYUKI

(54) NAVIGATION METHOD

(57)Abstract:

PROBLEM TO BE SOLVED: To improve convenience by presenting a logical structure related with resource arrangement, and making the expressing method suitable to the desire of a user.

SOLUTION: A navigation information address management table 301 is constituted of three fields, that is, a type 302, name 303, and data logical address 304. A tree type, tour type, or link type is described in the type. The leading address of a list structure is stored in the data logical address 304. Thus, navigation information in which the semantic arrangement structure of a resource is described can be automatically generated by a tree detecting processing, tour detecting processing, and link detecting processing for following a hyper link, and detecting the type of the information. The generated navigation information is transferred through a network, and stored so that the navigation information can be quickly referred to as a local file. Thus, the arrangement of plural resources is exactly displayed to a user so that the user can obtain the address of the desired resource.



LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

(19)日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開平10-222525

(43)公開日 平成10年(1998)8月21日

(51)Int.Cl.⁶

識別記号

F 1

G 0 6 F 17/30

G 0 6 F 15/403

3 7 0 Z

12/00

5 4 7

12/00

5 4 7 H

13/00

3 5 7

13/00

3 5 7 Z

審査請求 未請求 請求項の数6 O L (全 13 頁)

(21)出願番号 特願平9-24804

(22)出願日 平成9年(1997)2月7日

(71)出願人 000005108

株式会社日立製作所

東京都千代田区神田駿河台四丁目6番地

(72)発明者 山田 隆亮

神奈川県川崎市麻生区王禅寺1099番地 株

式会社日立製作所システム開発研究所内

(72)発明者 小嶋 弘行

神奈川県川崎市麻生区王禅寺1099番地 株

式会社日立製作所システム開発研究所内

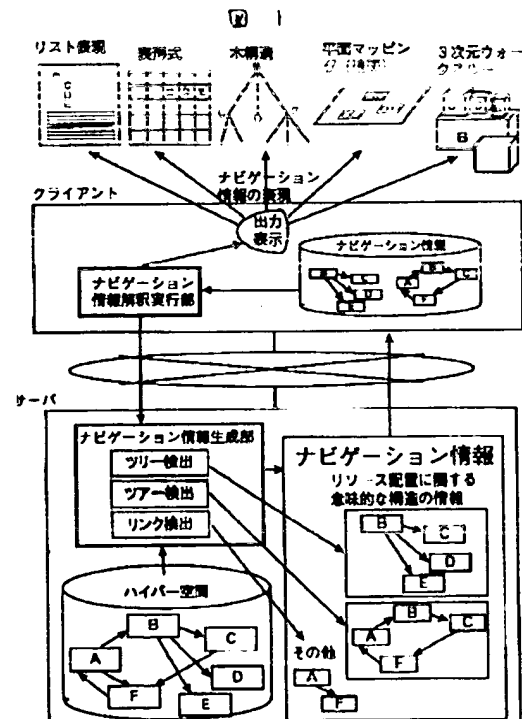
(74)代理人 弁理士 小川 勝男

(54)【発明の名称】 ナビゲーション方法

(57)【要約】

【課題】ナビゲーションに必要な情報をユーザに的確に提示することにより使い勝手を向上したハイパーテキストシステムを提供することにある。

【解決手段】ディスク装置、入力装置、出力装置を具備した計算機を複数相互接続した構成のネットワークハイパーテキストシステムにおいて、(1)ナビゲーションのためのリソース配置に関する論理構造を、その表現方法と分離して記述するナビゲーション情報の記述手法、(2)(1)のナビゲーション情報を自動生成する処理方法、(3)生成したナビゲーション情報をもつサーバから、ナビゲーション情報をダウンロードするネットワーク制御処理方法、(4)ダウンロードしてきたナビゲーション情報を解釈実行する処理方法から構成される。



【特許請求の範囲】

【請求項1】 ディスク装置1、入力装置1、出力装置1を具備したサーバと、ディスク装置2、入力装置2、出力装置2を具備したクライアントと、複数の該サーバと複数の該クライアントを相互接続するネットワークと、該ネットワークを介して該サーバのディスク装置1内の情報を、該クライアントのディスク装置2に転送する通信手段を具備したネットワーク型分散情報処理システムにおいて、下記のデータ、手段を具備し、(1)該サーバの該ディスク装置1内にあるリソースの配置に関するナビゲーション情報、(2)該サーバ上の該ナビゲーション情報を、該クライアントのディスク装置2にする通信手段、(3)該ナビゲーション情報内に記載されたリソース配置の論理的構造を解釈する手段、(4)解釈したナビゲーション情報を出力装置に出力する手段、該ナビゲーション情報は、リソース間の関係構造の記述をしたものであることを特徴とするナビゲーション方法。

【請求項2】 請求項1に記載したナビゲーション方法において、ナビゲーション情報(1)はアドレス管理テーブルを具備し、該アドレス管理テーブルは、ナビゲーションの型、名前、論理アドレスの先頭、を格納するフィールドを具備し、該論理アドレスの先頭に、リソースの配置をリスト構造に収めたデータの先頭アドレスを格納し、かつ、請求項1に記載したナビゲーション方法において、解釈したナビゲーション情報を下記5通りの方法、リソースの配置を3次元空間に表示して、ユーザにウォークスルーさせる方法、リソースの配置をグラフィカルに地図上にマッピングして表現する方法、リソースの配置を木構造に提示して、構造を見せる方法、リソースの配置を表にまとめて、表示する方法、リソースの配置をリスト状に羅列して提示する方法、で出力装置に出力する手段を具備したことを特徴とするナビゲーション方法。

【請求項3】 請求項1に記載したナビゲーション方法において、「ナビゲーション情報生成処理」の呼び出しを行なう第1のステップ；ナビゲーション情報転送処理を行なう第2のステップ；サブルーチン「ナビゲーション情報解釈実行処理」の呼び出しを行なう第3のステップ；を具備し、これらのステップを逐次実行制御する手段を具備したことを特徴とするナビゲーション方法。

【請求項4】 請求項3記載のナビゲーション方法において、ツリー型、ツア型、リンク型の3つのナビゲーションパターンを格納する手段を具備し、該ツリー型はリソース間の関係が、木構造の配置記述であり、該ツア型はリソース間のリンクが1方向に流れて1周する構造であり、該リンク型は該ツリー型でも該ツア型でもない、その他の構造であることを特徴とするナビゲーション方法。

【請求項5】 請求項4記載のナビゲーション方法において、リソースのリンク参照先がサーバ外である場合を区

別する手段を具備したことを特徴とするナビゲーション方法。

【請求項6】 請求項3記載のナビゲーション方法において、ナビゲーション情報を格納するファイルの用意を行なう第1のステップ；ツリー検出処理の呼び出しを行なう第2のステップ；ツア検出処理の呼び出しを行なう第3のステップ；リンク検出処理の呼び出しを行なう第4のステップ；ナビゲーション情報の格納を行なう第5のステップ；を具備し、これらのステップを逐次実行制御する「ナビゲーション情報生成処理」を具備したことを特徴とするナビゲーション方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】 本発明のナビゲーション方法は、ハイパーテキスト表現された文書処理システムに適用可能である。たとえば、WWW(World Wide Web)、電子マニュアルシステムや、事務連携管理システム、設計情報共有システム等に応用できる。

【0002】

【従来の技術】 本発明に係わる従来技術として、下記項目について順次説明する。

【0003】・HTMLファイル内での記述

- ・ナビゲーション問題
- ・インターネットロボット

(1) ファイル内での記述: WWW(World Wide Web) で用いられるHTML(Hyper Text Markup Language) は1つのリソース内の情報の論理構造表現のための記述言語であったが、複数リソースに対するナビゲーションのための情報を提供するには不備な点と冗長な点があった。ナビゲーションとは、リソースの配置(論理的相関関係)をユーザに的確に知らせることである。

【0004】 WWWでは、通常index.htmlと命名された特別なHTMLファイルを用いて1つのサイトを代表させている。しかし、1つのHTMLファイルに記載できる情報量ではサイトの全リソースをカバーできない。そこでリンクを使って、index.htmlと別のHTMLファイルを関係つけて見せる方法が用いられていた。ユーザにとっては、必要な情報が出てくるまでリンクを何度もたどる必要があった。その度に各階層のインデックスを記述したHTMLファイルをネットワーク越しに取り寄せる必要があった。

【0005】 HTMLは、リソースの内部の文章構造を記述するための言語であり、HTMLでナビゲーション目的にしたインデックス情報を記述するということは、ナビゲーションのためのリソース配置構造と、その表現形態を一緒に扱うことになる。このことは表示出力できる端末、ブラウザソフトウェアを限定することになり、好ましくない。リソース配置構造を表現するのに、3次元グラフィックスを用いることが必要なケースばかりではなく、文字情報のみの提示で十分なユーザも存在す

る。

【0006】(2) ナビゲーション問題: "J. C. Oaklin: "Hypertext: An Introduction and Survey", IEEE COMPUTER Sept., pp. 17-40 (1987)"に記載されたように、問題の根幹は、ハイパーテキストの迷子問題である。すなわち、ハイパー構造をユーザが把握しにくく、全体の中のどこを見ているかを見失うというハイパーテキストの構造的欠陥でもある。ハイパーテキスト構造に配置されたリソースにはナビゲーションが必須であり、現在のHTMLを用いた方法では、1度に送れるネットワーク転送速度がボトルネックとなって、適切なナビゲーション情報が提示されない。ハイパー構造のリソース(あるいはファイル)配置を計算機でディスプレイ表示するには、例えば、下記のような方法がある。

【0007】・リソースの配置を3次元空間に表示して、ユーザにウォークスルーさせる方法;
・リソースの配置をグラフィカルに地図上にマッピングして表現する方法;
・リソースの配置を木構造に提示して、構造を見せる方法;
・リソースの配置を表にまとめて、表示する方法;
・リソースの配置をリスト状に羅列して提示する方法;
しかし、リソースの配置に関する論理構造は、本来、その表現方法に関わりなく存在するものであるにも係わらず、HTML等既存の技術では混在して扱っている。例えば、HTMLファイルでは本来そのページに記述すべき情報の記述と、そのページに関係付けられたリンク情報とが記述されている。ナビゲーションにとっては、どのリンクがどこにつながっていて、どういう意味でつながれているのかが重要となる。すなわちリンク情報や、文章の構造情報は重要であるが、その中に記述された情報そのものは、ナビゲーションには余分なものであった。

【0008】(3) インターネットロボット
インターネットロボットは、WWW内の各サーバをめぐって、情報発見するためのソフトウェアツールである。インターネットロボットは、ナビゲーションの目的に用いられることはないが、本発明と類似した概念があるので、ここで述べる。

【0009】「Fah-Chun Cheong: "Internet Agents: Spiders, Wanderers, Brokers, and Bots", New Riders, Chapter 5, pp. 105-123 (1996)」によると、インターネットロボットは、ある一つのHTMLファイル内のリンク情報をもとに、そのページに関係付けられた別のHTMLファイルを自動的に取得する機能を具備する。この機能は再帰的に実行可能であり、理論的には世界中のWWWサーバの情報を取得可能である。収集した情報にインデックスを付けて、整理する機能を持つインターネットロボットも

ある。しかし、情報収集を目的としたインターネットロボットであるがゆえに、その整理方法は階層状態にまとめるものや、リストアップにとどめるものが多く、ナビゲーションのための論理構造としてまとめるものは存在しなかった。

【0010】複雑にリンクが張られてハイパー構造となっていることを考えずに、ハイパー構造を単純にツリー構造としてまとめるだけでは、情報欠落である。ツリーになっていない部分にこそリンク作成者の明確な意図が込められているにもかかわらず、切り捨てているのは、インターネットロボットの目的が情報収集を主目的としたもので、ナビゲーションの観点がなかったからである。

【0011】

【発明が解決しようとする課題】従来の技術においては、下記の課題があった。

【0012】第1にWWW等のハイパーテキストシステムにおいて、何度もリンクをたどらないと所望のリソース(ファイル)の場所がわからなかったため、使い勝手に問題があった。更にたどっている経過中のリソースは、情報転送にかかるネットワーク負荷を無意味に高めていた。

【0013】第2に、従来のナビゲーション機能では「論理構造の情報提供」という目的とその表現手段が一体化されていたため、ユーザは著者紹介等の、ナビゲーションに関わりの無い冗長な表現も一緒に受け取らざるをえず、遅い表示等を我慢していた。加えて、表現方法に係わる情報転送量の分、ネットワーク負荷を高めていた。

【0014】本発明の目的は、ナビゲーションに必要な情報をユーザに的確に提示することで使い勝手を向上し、ハイパーテキストシステムを提供することにある。例えば、ハイパーテキストシステムにおいて、リソース配置に関する論理構造を提示し、その表現方法はユーザの希望にあわせることのできる機能を提供することにある。

【0015】また、本発明のもう一つの目的は、表示方法と分離したナビゲーション情報記述方法の提供にある。

【0016】また、本発明の更にもう一つの目的は、上記ナビゲーション情報記述を自動的に行う機能の提供にある。

【0017】

【課題を解決するための手段】上記目的は次のようなナビゲーション方法の構成によって達成される。本発明の望ましい実施形態においては、ディスク装置、入力装置、出力装置を具備した計算機を複数相互接続した構成のネットワークハイパーテキストシステムにおいて、

(1) ナビゲーションのためのリソース配置に関する論理構造を、その表現方法と分離して記述するナビゲーション

ョン情報の記述手法;

(2) (1) のナビゲーション情報を自動生成する処理方法;

(3) 生成したナビゲーション情報をもつサーバから、ナビゲーション情報をダウンロードするネットワーク制御処理方法;

(4) ダウンロードしてきたナビゲーション情報を解釈実行する処理方法; から構成される。

【 0 0 1 8 】 (1) の記述手法として、ツリー型、ツアー型、リンク型を提起する。ツリー型はリソース間の関係が、木構造、あるいは階層構造であり、本の章立てや、リファレンスマニュアルのような配置記述を意味する。ツアー型は、ストーリー性をもった情報の列であり、リソース間のリンクが1方向に流れて1周する構造であり、作業手順書のような配置記述を意味する。最後のリンク型はツリー型でもツアー型でもない、いわばその他の構造である。リンク型は、リソース作者が配置記述する上で、参考程度にリンクをはったものと、明確にジャンプさせる意図を持たせたものの両極の意味を持つ。通常参考程度のリンクは別サーバに対して張られることが多く、サーバ名で両者を区別する手段を提供する。

【 0 0 1 9 】 本発明によると、ナビゲーションに必要なとなる情報は、サーバ内のリソース間の意味的な配置関係であり、ハイパーリンクで相互に関係付けられているリソース間の関係において、意味的な関係を持つものはツリー型、ツアー型、リンク型で大別しうる。

【 0 0 2 0 】 ハイパーリンクをたどり、これらの情報の型を検出するツリー検出処理、ツアー検出処理、リンク検出処理により、リソースの意味的な配置構造を記述したナビゲーション情報を自動的に生成できる。生成したナビゲーション情報をネットワークを介して伝送し、格納する処理により、ナビゲーション情報をローカルファイルとして、迅速に参照できる。ナビゲーション情報により、ユーザに複数リソースの配置を的確に表示することにより、ユーザは所望のリソースのアドレスを取得できる。その後、そのアドレスをもとに、ネットワーク経由でリソースを取得することができ、ネットワーク内を通るリソースは必要最小限に押さえることができる。

【 0 0 2 1 】

【 発明の実施の形態】 本発明の特徴の一つであるナビゲ

<

H I E R A R C H Y {

/;

/src/;

/src/arc.c;

/src/arc.h;

/src/backup/;

/src/makefile;

/src/backup/arc.c;

ーション情報の記述方法について、初めに説明する。次に、ナビゲーション情報の解釈実行処理、最後にナビゲーション情報の生成処理について、それぞれ発明の実施例を説明する。

【 0 0 2 2 】 図1 を使って、ナビゲーション情報を用いて、リソース配置を効率的に生成・解釈／実行するハイパーテキストシステムのシステム構成を詳細に説明する。

【 0 0 2 3 】 本実施例のハイパーテキストシステムはネットワークで相互接続されたサーバとクライアントから構成される。サーバはディスク装置、ネットワーク制御装置を具備した計算機である。サーバのディスク装置には、ハイパー空間内に配置されたリソースが格納されている。サーバの計算機メモリ内にはナビゲーション情報生成処理方法を示した処理手順がある。クライアントは、ディスク装置、マウス、ディスプレイ、ネットワーク制御装置を具備した計算機である。

【 0 0 2 4 】 このハイパーテキストシステムにおいて、ナビゲーション機能を利用する手順を以下で説明する。まず、クライアント要求により、サーバで、サブルーチンナビゲーション情報生成処理の呼びだしを行なう。ナビゲーション情報生成処理により、ナビゲーション情報ファイルが生成される。(ナビゲーション情報ファイルはリソース配置に関する意味的な情報の構造を記述したファイルであり、後で説明する。) 次に、サーバからネットワークを介してクライアントへ、ナビゲーション情報ファイルの転送を行なう。最後にクライアントで、サブルーチンナビゲーション情報解釈実行処理の呼びだしを行なう。ナビゲーション情報解釈実行処理により、ユーザの希望にあわせて、下記のナビゲーション方法のいづれかを、クライアントのディスプレイに出力する。

【 0 0 2 5 】 ・リソースの配置を3次元空間に表示して、ユーザにワークスルーさせる方法;

・リソースの配置をグラフィカルに地図上にマッピングして表現する方法;

・リソースの配置を木構造に提示して、構造を見せる方法;

・リソースの配置を表にまとめて、表示する方法;

・リソースの配置をリスト状に羅列して提示する方法;

ナビゲーション情報ファイルの記述例を下記に示す。

【 0 0 2 6 】

```

    };
    SCENARIO {
        /another/q1.html
        /another/q2.html
        /another/q2-1.html;
        /another/q2-2.html;
    };
    /another/q3.html;
};
TOUR { /another/q4.html; /another/q5.html; /another/q6.html; };
>

```

まず、ナビゲーション情報の制御文字として、下記の文字を定義する。

【0027】・ナビゲーション情報の始まりを示す記号「<」

・ナビゲーション情報の終わりを示す記号「>」

・一連の情報のブロックの始まりを示す記号「{」

・一連の情報のブロックの終わりを示す記号「}」 20

・1つの情報単位の終わりを示す記号「;」

最小の情報単位である「1つのリソース」の場所を示す情報としてそのアドレスを例えば、「resource-address;」のように記する。ここでいうリソースは、例えばHTMLファイルのような、ナビゲーション対象となる最小情報単位をいうものとする。逆にナビゲーションとは、リソースの置き場所をユーザに提示することであ

り、ナビゲーション情報とは、リソース間の論理的関係を示したものである。

【0028】また、複数のリソースをまとめた1固まりの情報ブロックは、「{」と「}」を用いて、複数のリソースを列記する。また、情報ブロックの中に、再度情報ブロックを定義できるように括弧のネストを可能とする。

【0029】具体のナビゲーション情報の文法記述例として、sample.1とsample.yを下記に示す。sample.1は上記制御文字、<、>、{、}、;を定義する。文法の厳密な説明については、例えば、「五月女: "yacc/lex", 啓学出版, 1992」の記法を参照されたい。

【0030】

```

sample.1 : -----
%%
<      { return START; }
>      { return STOP; }
{      { return LEFT; }
}      { return RIGHT; }
[a-zA-Z]+ { return WORD; }

```

sample.yは、sample.1で定義したSTART, STOP, END, LEFT, RIGHT, WORDの6つをトークンとした文法記述を定義している。文法の厳密

な説明については、例えば、「五月女: "yacc/lex", 啓学出版, 1992」の記法を参照されたい。

【0031】

```

sample.y : -----
%token START STOP END LEFT RIGHT WORD
%%
navi      : START blocks STOP
blocks    : block END
           | blocks block END
;
block     : WORD LEFT resources RIGHT END
;
resources : WORD END
           | resources WORD END
           | WORD LEFT resources RIGHT END
;

```

%%

「五月女: " yacc/lex" 啓学出版, 1992」によると、上記定義により、この文法規則に則ったパーザが、容易に生成可能である。パーザは、定義した文法規則に則った情報を抜き出してくるプログラムであり、本パーザでは、情報ブロックの名称と、その情報ブロックを構成するリソースを抜き出す処理を行う。以降で説明する処理方法は、ここで示したパーザにより抽出した情報をもとに処理を行うものである。

【0032】更に、情報ブロックの名称に、次に示す特別な名前を定義する。

【0033】・HI RARCHY
・SCENARIO
・TOUR

HI RARCHYはリソースが階層構造に配置されている情報ブロックを示す。SCENARIOはハイパーリンクによりリソースが包含関係に互いに関係付けられた情報ブロックであり、次に示すTOURでないものを示す。TOURは、SCENARIOのようにリソースが互に関係付けられている情報ブロックであるが、関係付けが下記に示す一定の規則に則ったものを示す。一つの説明内容をいくつかのリソースに分けて表現したものであり、ハイパーリンクの参照関係が一つのツアーのように、決ったルートを通して一連のストーリーを展開す

```
typedef struct __list{
    __list *parent;
    __list *children;
    __list *left, *right;
    int resource_address;
} *hirarchy_list, *link_list, *tour_list;
```

図3を使って、ナビゲーション情報格納形式を詳細に説明する。

【0037】ナビゲーション情報アドレス管理テーブル301は、型302、名前303、データ論理アドレス304の3つのフィールドから構成する。型には、ツリー型か、ツアー型か、リンク型かを記述する。データ論理アドレス304には、図2で説明したリスト構造の先頭アドレスを格納する。

【0038】図4を使って、ナビゲーションパターンの概念を詳細に説明する。

【0039】ナビゲーション情報の記述手法として提起した、ツリー型、ツアー型、リンク型のそれぞれの例を示す。ツリー型はリソース間の関係が、木構造、あるいは階層構造であり、本の章立てや、リファレンスマニュアルのような配置記述を意味する。ツアー型は、ストーリー性をもった情報の列であり、リソース間のリンクが1方向に流れて1周する構造であり、作業手順書のような配置記述を意味する。最後のリンク型はツリー型でもツアー型でもない、いわばその他の構造である。リンク型は、リソース作者が配置記述する上で、参考程度にリ

*る構造のものをTOURとする。HI RARCHY、SCENARIO、TOURのデータ構造、処理方法については後で図を用いて詳細に説明する。

【0034】図2を使って、ナビゲーション情報のデータ形式を詳細に説明する。

【0035】ナビゲーション情報は、複数のリソース、情報ブロックが複雑に関係しあう状態を表現するため、情報格納形式は、追加・削除のしやすいリスト構造を用いる。ナビゲーション情報ファイルを計算機メモリに格納するとき、本実施例では、リスト構造は、ひとつのノードを親とみなしたとき、それと関係する子ノードへのポインタと、自分の親のポインタ、自分の兄、弟へのポインタの4つのポインタと、どのリソースに関係するかを示すリソースアドレスへのポインタを具備する。このリスト構造においては、親は自分の子供へのポインタは1つしか持たないため、長男のポインタだけをもつ。長男は自分の兄弟へのポインタを持っているため、親は長男のポインタを介して自分の子供へのポインタを知ることができる。このようなデータ構造により、ハイパーリンクのような指向性のあるナビゲーション情報を柔軟に記述できる。言語的に記述すると、下記のような構造である。

【0036】

リンクをはったものと、明確にジャンプさせる意図を持たせたものの両極の意味を持つ。適宜参考程度のリンクは別のリーバに割当てられることが多く、リーバ名で両者を区別する手段を提供する。

【0040】図4は、サーバ内にA～Fのリソースがハイパー空間に配置されている状態である。ツリー検出処理、ツアー検出処理、リンク検出処理により、Bを中心に考えた場合のナビゲーション情報を抽出する。これらの処理内容については、以下で図10以下を用いて詳細に説明する。

【0041】図5を使って、まずナビゲーション情報解釈実行処理を詳細に説明する。

【0042】まず、処理5001で、前処理を行なう。すなわち、内部で用いる変数の初期化と、ユーザからの出力指示要求を受ける。次に、処理5002で、ナビゲーション情報を格納したファイルの読み込みを行なう。次に、処理5003で、ナビゲーション情報が続く間、処理5004で場合分けを行なう。場合分け処理5004で、ブロック情報がHI RARCHYの時には、処理5005で、階層制御処理を行なう。場合分け処理50

04 で、ブロック情報がSCENAR! Oの時には、サブルーチン リンク制御部の呼びだしを行なう。場合分け処理5004で、ブロック情報がTOURの時には、処理5007で、サブルーチン ツアー制御部の呼びだしを行なう。次に、処理5008で、ナビゲーション情報を格納したファイルを閉じる。最後に、処理5009で、後処理を行なう。すなわち、ユーザ要求に基づいて、出力する方法を選択する。論理構造を把握した後の、個別の出力方法自体は従来の技術であり、下記の方法を提供する。

【0043】・リソースの配置を3次元空間に表示して、ユーザにウォークスルーさせる方法;
・リソースの配置をグラフィカルに地図上にマッピングして表現する方法;
・リソースの配置を木構造に提示して、構造を見せる方法;
・リソースの配置を表にまとめて、表示する方法;
・リソースの配置をリスト状に羅列して提示する方法;
図6を使って、ツリー制御処理の処理を詳細に説明する。

【0044】まず、処理6001で、前処理を行なう。ナビゲーション情報アドレス管理テーブル301のオープン等である。次に、処理6002で、図2で説明したリスト構造のメモリ領域の確保、初期化を行なう。次に、処理6003で、階層部分の行がなくなるまでの間、処理6004以下の一連の処理を繰り返す。まず、処理6004で、新規リストのメモリ領域生成を行なう。次に、条件判定処理6005で、ディレクトリを調べる。ディレクトリ内には、再度ディレクトリがあるか、ファイルがあるかのいずれかである。ここで、ディレクトリを見つけたときは、処理6006以下の一連の処理を行なう。まず、処理6006で、新規リストヘディレクトリ名の登録を行なう。次に、処理6007で、新規リストのホスト末尾への接続を行なう。判定処理6005でファイルを見つけたときには、処理6008以下の一連の処理を行なう。まず、処理6008で、新規リストヘファイル名の登録を行なう。次に、処理6009で、既存ディレクトリ名の所へ新規リスト追加を行なう。

【0045】最後に、処理6010で、後処理を行なう。ナビゲーション情報アドレス管理テーブルの更新である。型は全てツリー型であり、名前については、各リソースから適切な名前を抽出する。例えば、HTMLファイルのヘッダ部分にはタイトルが記述されているので、これを用いることができる。データ論理アドレスには、リストへのポインタを格納する。

【0046】図7を使って、リンク制御処理を詳細に説明する。

【0047】まず、処理7001で、前処理を行なう。図6の前処理6001とほぼ同等の処理である。次に、

処理7002で、リスト構造のメモリ領域の確保、初期化を行なう。次に、処理7003で、シナリオ部分の行がなくなるまでの間、処理7004で、サブルーチン リンクの再帰処理の呼びだしを行なう。最後に、処理7005で、後処理を行なう。図6の後処理6010とほぼ同等の処理である。短い主処理であるが、再帰処理は主処理と分離して記述しないと再帰的に実行させられない。このことは以下の処理フローでも同様である。

【0048】図8を使って、リンクの再帰処理の処理を詳細に説明する。

【0049】まず、処理8001で、新規リストのメモリ領域生成を行なう。最後に、条件判定処理8002で、ブロック名を調べ、この条件が成立するときは、処理8003以下の一連の処理を行なう。まず、処理8003で、新規リストの登録を行なう。次に、処理8004で、既存リストへの接続を行なう。次に、処理8005で、サブルーチン リンクの再帰処理の呼びだしを行なう。判定処理8002で条件が成立しないときには、処理8006以下の一連の処理を行なう。まず、処理8006で、新規リストの登録を行なう。次に、処理8007で、既存リストへの接続を行なう。

【0050】図9を使って、ツアー制御処理を詳細に説明する。

【0051】処理9001で、シナリオ部分の行がなくなるまでの間、処理9002以下の一連の処理を繰り返す。まず、処理9002で、新規リストの登録を行なう。次に、処理9003で、既存リストへの接続を行なう。なお、ツリー制御処理やツリー制御処理と同様の前処理、後処理も実行する。

【0052】図10を使って、ナビゲーション情報生成部の処理を詳細に説明する。

【0053】まず、処理10001で、前処理を行なう。内部変数の初期化等である。次に、処理10002で、ナビゲーション情報を格納するファイルの用意を行なう。次に、処理10003で、サブルーチン 階層情報生成部の呼びだしを行なう。次に、処理10004で、サブルーチン リンク情報生成部の呼びだしを行なう。次に、処理10005で、サブルーチン ツアー情報生成部の呼びだしを行なう。次に、処理10006で、ナビゲーション情報の格納を行なう。最後に、処理10007で、後処理を行なう。内部変数の開放等である。

【0054】図11を使って、ツリー検出処理を詳細に説明する。

【0055】まず、処理11001で、ルートディレクトリのセットを行なう。最後に、処理11002で、サブルーチン ツリー構造の再帰処理の呼びだしを行なう。ツリー構造の再帰処理の自己呼び出し処理により、ナビゲーション情報ファイルにツリー情報が登録される。

10

20

30

40

50

13

【0056】図12を使って、ツリー構造検出の再帰処理を詳細に説明する。

【0057】まず、処理12001で、セットされたディレクトリの情報抽出を行なう。最後に、処理12002で、ディレクトリ情報がなくなるまでの間、条件判定処理12003で、サブディレクトリを調べ、ディレクトリをみつけた場合、処理12004以下の一連の処理を行なう。まず、処理12004で、サブディレクトリのセットを行なう。次に、処理12005で、サブルーチン ツリー構造検出の再帰処理の呼び出しを行なう。判定処理12003でファイルを見つけたときには、処理12006で、ファイル名の登録を行なう。その書式は、ナビゲーション情報ファイルに従う。

【0058】図13を使って、リンク検出処理を詳細に説明する。

【0059】まず、処理13001で、メモリ内にリスト領域の新規作成を行なう。次に、処理13002で、ウェルカムページのセットを行なう。ウェルカムページは、ハイパー構造のリソースにとっての、ルートディレクトリに相当する。最後に、処理13003で、サブルーチン ハイパー構造の再帰処理の呼び出しを行なう。

【0060】図14を使って、リンク構造検出の再帰処理を詳細に説明する。

【0061】まず、処理14001で、セットされたページのリンク情報抽出を行なう。最後に、処理14002で、リンク情報がなくなるまでの間、処理14003以下の一連の処理を繰り返す。まず、処理14003で、リンク先の登録を行なう。次に、処理14004で、リンク先のページをセットを行なう。次に、処理14005で、サブルーチン ハイパー構造の再帰処理の呼び出しを行なう。

【0062】図15を使って、ツアー情報生成部の処理を詳細に説明する。

【0063】まず、処理15001で、ウェルカムページのセットを行なう。最後に、処理15002で、サブルーチン ツアー検出の再帰処理の呼び出しを行なう。

【0064】図16を使って、ツアー検出の再帰処理の処理を詳細に説明する。

【0065】まず、処理16001で、全ての子供について調べ終るまでの間、処理16002以下の一連の処理を繰り返す。まず、条件判定処理16002で、子供同士で親子関係があるかどうかを調べ、この条件が成立するときは、処理16003で、子供のアドレスの記憶を行なう。次に、処理16004で、子供のページのセットを行なう。次に、処理16005で、サブルーチン ツアー検出の再帰処理の呼び出しを行なう。最後に、

14

処理16006で、親と記憶した子供のアドレスのツアー登録を行なう。

【0066】

【発明の効果】第1に本発明によれば、WWW等のハイパーテキストシステムにおいて、何度もリンクをたどらないと所望のリソース(ファイル)の場所がわからなかったのと比較して、全体の構造が把握できる。最終的に必要な文書のみ取得することで、情報転送にかかるネットワーク負荷の低減ができる。

【0067】第2に、従来のナビゲーション機能では「論理構造の情報提供」という目的とその表現手段が一体化されていたため、ユーザは冗長な表現も一緒に受け取らざるをえず、違い表示等を我慢していた。本発明によると、論理構造の情報のみを受け取り、その表現手段をユーザに託すことができる。その結果、ユーザの利用状況に適応した形でのナビゲーションが可能となり、使い勝手が格段に向上する。同時に、クライアント側で、ナビゲーションのためのインタフェース処理を行うので、ネットワーク負荷の低減ができる。

【0068】第3に、本発明によると、上記の効果を生むナビゲーション情報を自動生成することができる。ナビゲーションのための情報を生成する労力の省力化が可能である。

【図面の簡単な説明】

【図1】システム構成の説明図である。

【図2】ナビゲーション情報格納形式の説明図である。

【図3】ナビゲーション情報アドレス管理テーブルの説明図である。

【図4】ナビゲーションパターン概念図の説明図である。

【図5】ナビゲーション情報解釈実行処理の説明図である。

【図6】ツアー制御処理の説明図である。

【図7】リンク制御処理の説明図である。

【図8】リンク構造の再帰処理の説明図である。

【図9】ツアー制御処理の説明図である。

【図10】ナビゲーション情報生成処理の説明図である。

【図11】ツリー情報検出処理の説明図である。

【図12】ツリー構造の再帰処理の説明図である。

【図13】リンク情報検出処理の説明図である。

【図14】リンク構造検出の再帰処理の説明図である。

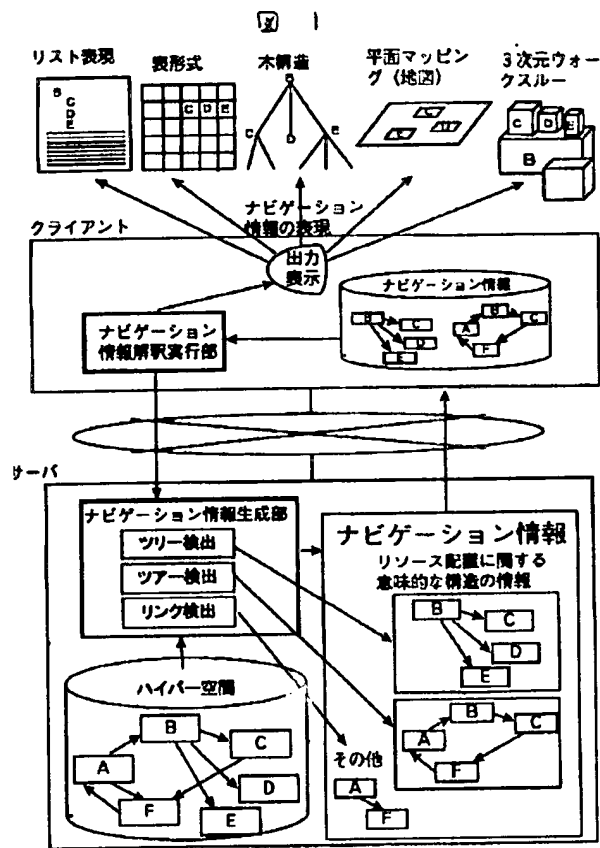
【図15】ツアー情報生成部の説明図である。

【図16】ツアー構造検出の再帰処理の説明図である。

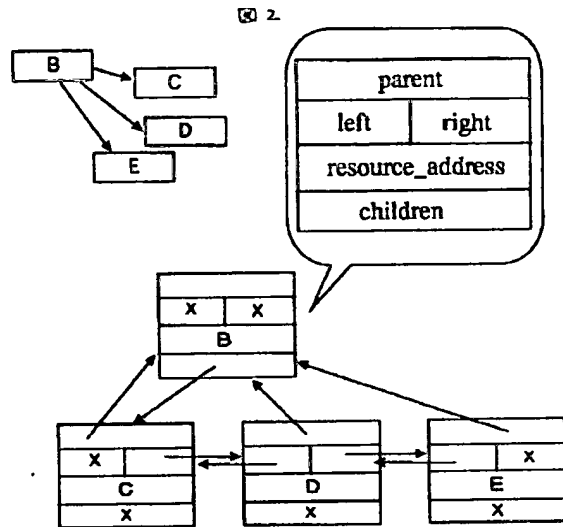
【符号の説明】

301…ナビゲーション情報。

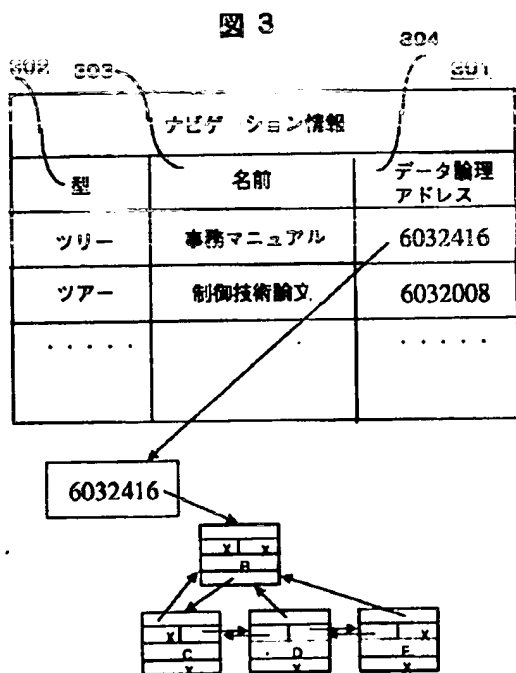
【 図1 】



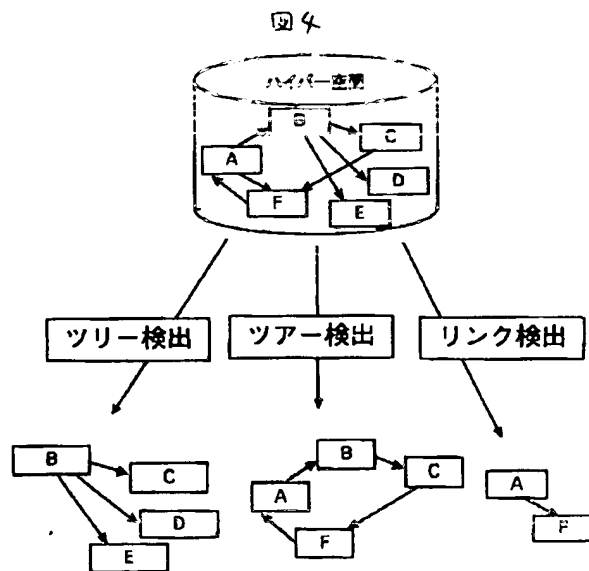
【 図2 】



【 図3 】

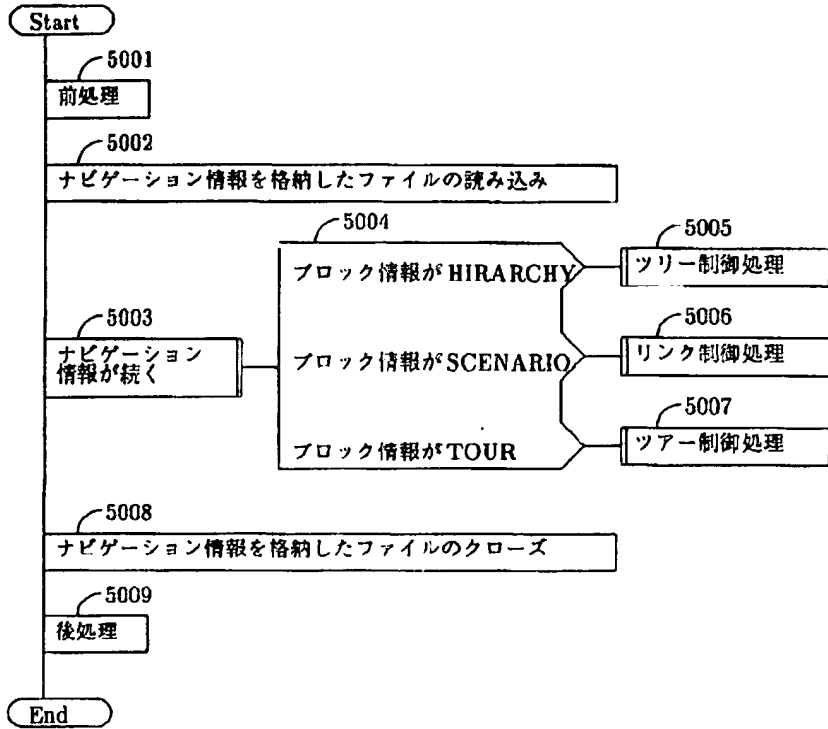


【 図4 】



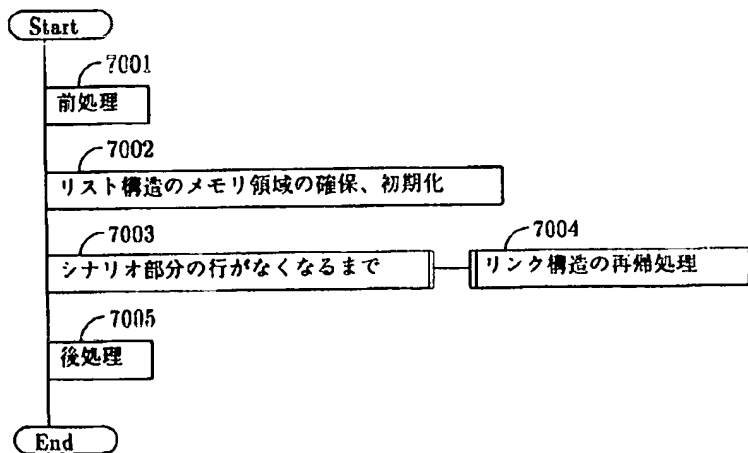
【 図5 】

図 5



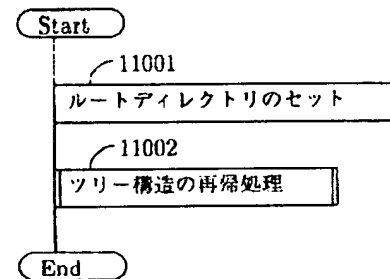
【 図7 】

図 7



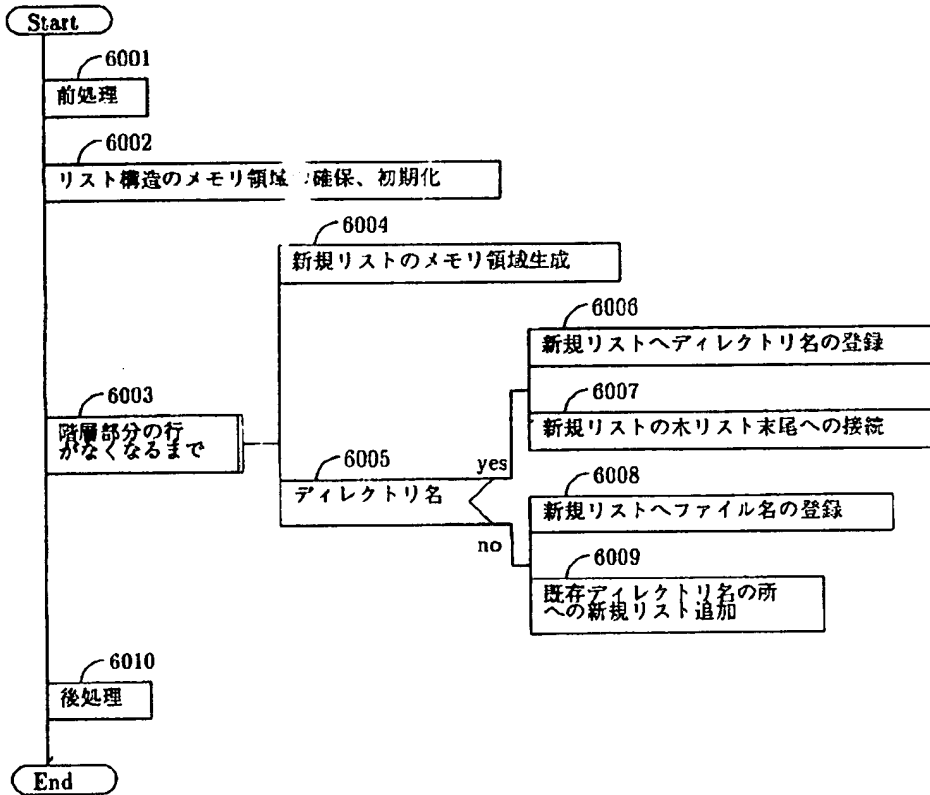
【 図11 】

図 11



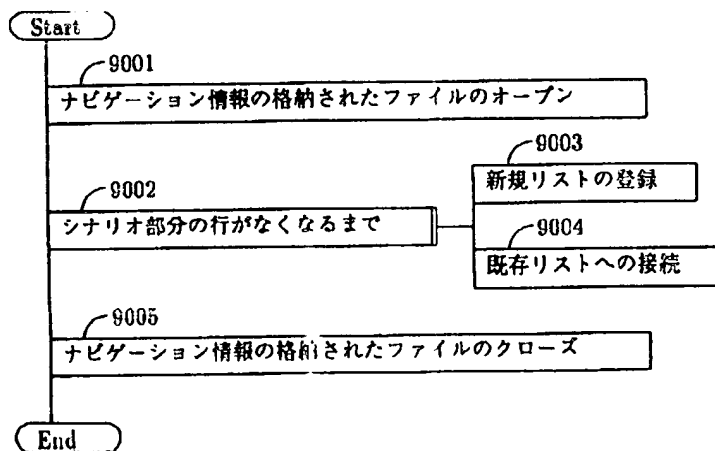
【 図6 】

図 6



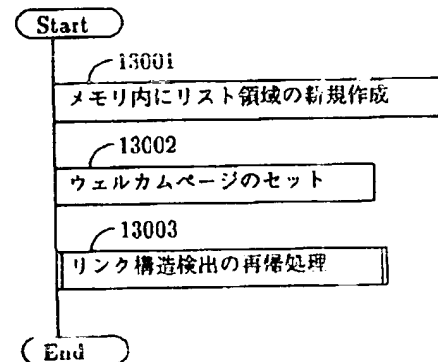
【 図9 】

図 9



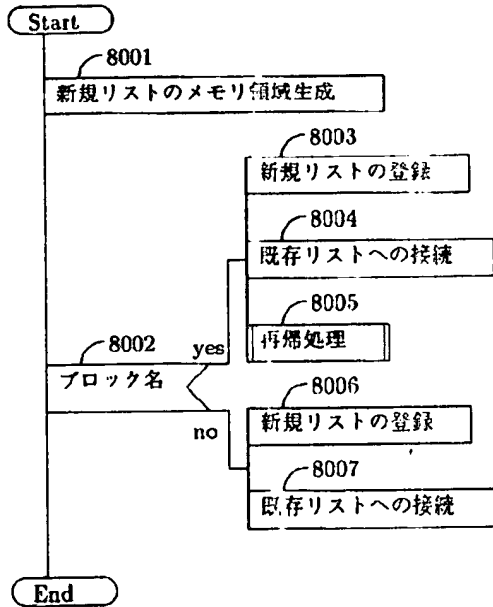
【 図13 】

図 13



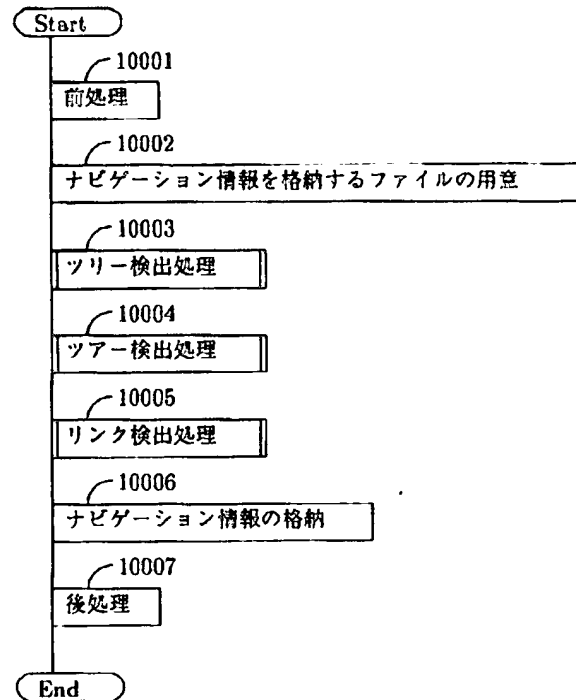
【 図8 】

図 8



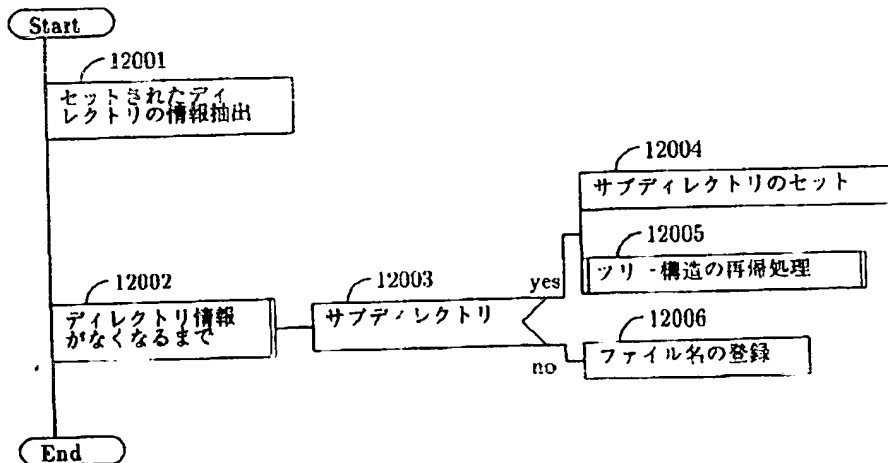
【 図10 】

図 10



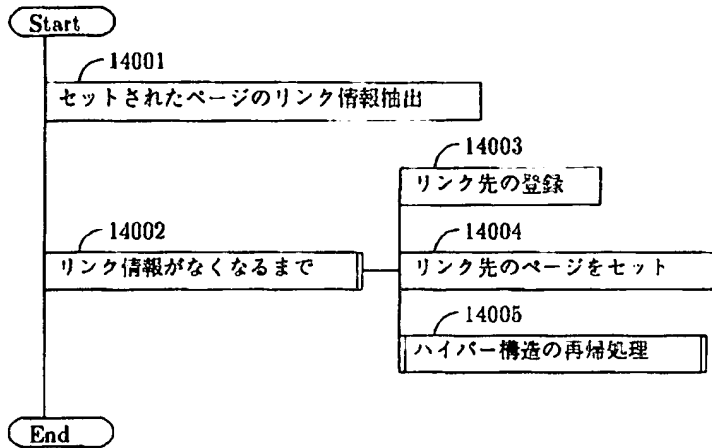
【 図12 】

図 12



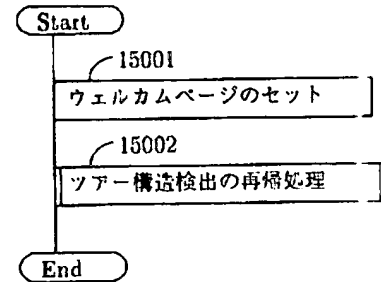
【 図14 】

図 14



【 図15 】

図 15



【 図16 】

図 16

